心得总结（第三章）

第三章的学习是围绕着“类“来出的，类是 C++ 的核心特性，在C++中，除了已有的数据类型（整型，字符型）外，还可以自定义数据类型，这种类型通常被称为用户定义的类型。类用于指定对象的形式，它包含了数据表示法（数据类）和用于处理数据（操作类）的方法。类中的数据和方法称为类的成员。函数在一个类中被称为类的成员。类定义是以关键字 class 开头，后跟类的名称。类的主体是包含在一对花括号中。类定义后必须跟着一个分号或一个声明列表。例如，我们使用关键字 class 定义 Box 数据类型，如下所示：

class Box

{

public://公有成员 在类外可以访问

double length; // 盒子的长度

double breadth; // 盒子的宽度

double height; // 盒子的高度

};

① 如果类的定义和主函数在同一个源文件里，那么就会可能遇到这样的问题：在类定义之 前，主函数使用了这个类。这将会导致错误的发生，因为主函数还没有意识到这个类的存在。 所以必须在主函数之前声明这个类的存在。

② 还可以在头文件中定义类，然后在.Cpp源文件中包含这个头文件，由于包含头文件在主函数之前，所以不用在主函数中声明这个类。

Student.h //头文件

class Student //学生类的定义

{

……

};

main.cpp //文件

#include "Student.h" //要注意这里必须用双引号，而不能用尖括号

int main()

{

……

}

声明一个对象的引用方法是：        类名 &对象名 a=对象名 b;对对象 a 的访问和操作就如同对对象 b 的访问和操作一样，对象 a 只是对象 b的一个“绰号”。

友元函数；友元使设计程序方便了很多。原先的那些私有成员都能轻松地被访问了。于是不用去写那些繁琐的成员函数，程序执行的时候也减少了函数的调用次数，提高了运行效率。友元的存在，破坏了类的封装性。一个类出现问题，就不仅仅是由这个类本身负责了，还可能和它众多的友元有关